

**RETIRED AVAILABLE COPY**

02/18 '18 05:10 NO.448 02

**Cite No. 17.**

Abstract of CN1337033/Publication date- 2002.2.20

Abstract of corresponding document: US2002085016

A method and apparatus for image scaling is provided. A 3D pipeline comprises a command stream controller to enable a rectangle mode. The 3D pipeline including a windower to produce addresses for the rectangle defined by the vertices. A filter interpolates between neighboring points, based on relative location, to generate attributes for each pixel. A color calculator aligns output data and writes the output data to a destination surface.

[19] 中华人民共和国国家知识产权局

[51] Int. Cl<sup>7</sup>

G06T 3/40

## [12] 发明专利申请公开说明书

[21] 申请号 00802788.9

[43] 公开日 2002 年 2 月 20 日

[11] 公开号 CN 1337033A

[22] 申请日 2000.1.12 [21] 申请号 00802788.9

[30] 优先权

[32] 1999.1.15 [33] US [31] 09/231,260

[86] 国际申请 PCT/US00/00784 2000.1.12

[87] 国际公布 WO00/42571 英 2000.7.20

[85] 进入国家阶段日期 2001.7.13

[71] 申请人 英特尔公司

地址 美国加利福尼亚州

[72] 发明人 V·G·摩克 S·W·詹森

[74] 专利代理机构 中国专利代理(香港)有限公司

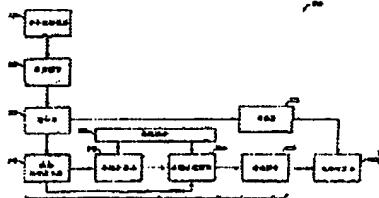
代理人 吴增勇 傅康

权利要求书 3 页 说明书 10 页 附图页数 6 页

[54] 发明名称 使用三维流水线伸拉位图的方法和装置

[57] 摘要

提供一种用于图像缩放的方法和装置。3D 流水线包括命令流控制器以启动矩形方式。所述 3D 流水线包括窗口器以产生用于由顶点定义的矩形的地址。滤波器内插在基于相对位置的相邻点之间，以产生每个像素的属性。色彩计算器调整输出数据并把输出数据写到目标表面。



ISSN 1008-4274

知识产权出版社出版

01-07-13

## 权 利 要 求 书

1. 一种用于三维着色和伸拉位图的 3D 流水线，所述用于伸拉位图的 3D 流水线包括：
  - 5 用于启动矩形方式的命令流控制器；
  - 用于产生由顶点定义的矩形的地址的窗口器；
  - 在基于相对位置的相邻点之间插入的、用于每个像素以产生每个像素的属性的滤波器；
  - 10 用于校准输出数据并且把所述输出数据写到目标表面的色彩计算器。
2. 权利要求 1 的 3D 流水线，其特征在于还包括：设置引擎，它计算关于 X 和 Y 及原点的属性变化率、产生源和目标地址的坐标比、并产生由传递给所述窗口器的所述矩形定义的有边界框的。
- 15 3. 权利要求 1 的 3D 流水线，其特征在于还包括：产生在像素映射中提取特定像素的实际地址的映射地址产生器。
4. 权利要求 3 的 3D 流水线，其特征在于，所述映射地址产生器利用顶点的非归一化值。
5. 权利要求 1 的 3D 流水线，其特征在于还包括：20 用于提供存储管理和缓存处理的读取单元。
6. 权利要求 1 的 3D 流水线，其特征在于还包括：用于执行 YUV 空间中的通道转换的通道操作单元。
7. 权利要求 6 的 3D 流水线，其特征在于：所述转换是从 4:2:2 YUV 空间到 4:2:0 YUV 空间。
- 25 8. 权利要求 1 的 3D 流水线，其特征在于：所述色彩计算器还要执行彩色空间转换。
9. 权利要求 8 的 3D 流水线，其特征在于：所述彩色空间转换是从 YUV 到 RGB。

01-07-13

10. 权利要求 1 的 3D 流水线，其特征在于还包括：  
用于产生  $\alpha$  混合的  $\alpha$  值的内插器。
11. 权利要求 10 的 3D 流水线，其特征在于：所述色彩计算器还用于使用由所述内插器提供的所述  $\alpha$  值对所述输出数据进行  $\alpha$  混合。
- 5 12. 权利要求 1 的 3D 流水线，其特征在于：所述窗口器产生用于目标表面的地址。
13. 权利要求 12 的 3D 流水线，其特征在于：所述目标表面是强制通过对整个有边界框的边缘测试而产生的。
14. 权利要求 1 的 3D 流水线，其特征在于：所述窗口器用深层流 10 水线作业以持续产生地址、并且随着产生地址而把所述地址持续地发送到存储器，由此减小存储器延迟的影响。
15. 一种在三维(3D)流水线中的方法，它包括：  
接收信号；  
接收定义一个区域的两个或两个以上坐标点；  
15 基于所述信号确定所述区域是三角形还是矩形；  
如果所述区域是三角形则基于关于所述区域的数据而产生 3D 图像；和  
如果所述区域是矩形，则对关于所述区域的数据执行伸拉位图操作。
- 20 16. 权利要求 15 的方法，其特征在于还包括：  
基于所述信号设置所述 3D 流水线的状态，所述状态表示所述坐标点是定义视频图像还是定义用于着色的三维图像。
17. 权利要求 15 的方法，其特征在于还包括：  
如果所述区域是矩形，则在处理所述数据之前刷新所述 3D 流水 25 线。
18. 权利要求 15 的方法，其特征在于：所述信号是矩形命令。
19. 权利要求 15 的方法，其特征在于：所述多个坐标点包括定义直角三角形的左上顶点、右下顶点和左下顶点的三个点，通过确定右

01-07-13

上顶点来定义所述矩形。

20. 一种在三维(3D)流水线中对数字视频数据进行伸拉位图的方法，所述方法包括：

- 接收矩形命令；
- 5 接收定义直角三角形的左上顶点、右下顶点和左下顶点；
- 标识定义矩形的有边界框；
- 计算图像目标表面；
- 计算图像源表面；
- 抽取和缓存像素值；
- 10 内插所述像素值以产生在所述目标表面中的像素； 和
- 显示所述目标表面。

01-07-13

## 说 明 书

### 使用三维流水线伸拉位图的方法和装置

#### 5 发明领域

本发明涉及电子装置的图形显示。更具体地说，本发明涉及对由电子装置显示的图形的伸拉位图(stretch blitting)操作。

#### 技术背景

10 经过二十多年个人计算机已经走入了日常家用。用于教育和娱乐的游戏已经有用于 PC 的强大的应用软件并且由于三维(3D)图示技术的引入而得以增强。特定纹理的实时图片 3D 图示技术将提供对具有全视点自由度、可见到的真实世界位置的准确模拟。

15 高质量实时 3D 图示技术的关键是，要具有支持杂乱场景的着色的系统结构和存储器访问方案；在所述杂乱场景中有抬高了像素存储器访问次数的多层对象。相关的处理负荷必须不会把系统刷新率降低到我们认为的实时(30Hz)以下。一些提出的解决方案把处理卸载给应用程序(application's shoulders)，因而在对简单场景进行实时着色时有困难。更好的加速器将加速那些处理器增强的硬件功能，留着处理器用于高级功能。一般，定制的加速硬件(专用集成电路 ASIC - 硅)执行着色功能更快，为软件应用程序留有更多时间。这种图形 20 3D 着色引擎，即 3D 流水线，使用大量的乘法和加法电路。

25 电视会议和其他视频图像处理也提到日程中。一个期望用在可视电信会议中的特征是在传输前把视频图像缩放到各种大小的能力。图像越小、同时提供的细节越少，则需要的通信带宽越小并且能被接收者计算机越快地再现。一般，摄相机将产生其最大尺寸由摄相机图像传感器限定的视频图像。然后可由缩放装置把该图像缩放成更小或更大的图像。

01-07-13

5

采用对源像素的进行数字滤波来提供缩放的输出的现有的视频  
摄相机被设计成在源像素时钟的单个周期内产生每个输出像素。换  
言之，多抽头滤波器的抽头中的每个必须用在同一个时钟周期中。  
因此，需要大量乘法和加法电路来并行处理每个单独的滤波器抽头。  
随着乘法和加法电路的数量增加，滤波器传播延迟和功耗也增加。  
而且，当缩放装置是用集成电路(IC)来实现时，加法和乘法电路的增  
加的数量转变为增加的实现 IC 的芯片尺寸。最终结果是更昂贵的缩  
放引擎。

10

如果计算机系统包括 3D 着色以及视频成像系统，则用于辅助加  
速计算的硬件就面积和门而论变得昂贵。因而，如果减少用于包括  
三维着色和视频缩放的成像系统的门数量，则会有好处。

20

#### 发明概述

15

本发明提供用于图像缩放的方法和装置。3D 流水线包括启动矩  
形模式的命令流控制器。所述 3D 流水线包括产生由各顶点定义的矩  
形的地址的窗口器(windower)。在基于相对位置的相邻点之间插入滤  
波器，以产生每个像素的属性。色彩计算器调整输出数据并且把输出  
数据写到目标表面。

#### 附图简介

在附图中，以举例的方式、而不是以限定的方式说明本发明，  
并且在附图中类似的标号表示类似的元件，其中：

25

图 1 是本发明可以在其中实现的计算机系统的方框图；

图 2 是用于处理视频图像的 3D 流水线的一个实施例的方框图；

图 3 是视频图像伸拉处理的一个实施例的流程图；

图 4 说明用于 16 比 12 的矩形的目标数据；

图 5A 说明由用于 3D 着色和用于伸拉位图的窗口器执行的边缘  
测试；

图 5B 说明由用于 3D 着色和用于伸拉位图的窗口器执行的边缘

01-07-13

测试。

#### 详细描述

现在描述用于以纹理映射引擎进行图示技术的伸拉位图的方法和装置。在以下描述中，为说明起见，给出大量特定细节以便提供对本发明的全面理解。但是显然，对于本专业的技术人员，可以不用这些特定的细节来实践本发明。另外，以方框图的形式表示结构和装置以避免使本发明模糊不清。

在说明书中提到“一个实施例”或“实施例”指的是联系实施例描述的具体特征、结构或特性是被包括在本发明的至少一个实施例中的。“在一个实施例中”这个短语在说明书中各个位置的出现不一定都指的是同一实施例。

在包括三维着色和视频的计算机系统中，实现重复使用辅助 3D 着色和伸拉位图的硬件所用的门的系统。用于伸拉位图和 3D 着色的计算是类似的。因而，可以使用带有修改的、用于对三维图像着色的 3D 流水线来对视频图像放大和缩小、滤波、改变表面格式，以及改变彩色空间。如下所述，在 3D 流水线中用纹理映射引擎来进行图像处理。这是有好处的，因为用于修改 3D 流水线以允许图像处理的门数量，与单独地实现这些功能相比，大大地减少了。

采用 3D 流水线来执行伸拉位图导致性能改善、并且节省了孤立的伸拉位图引擎所需要的门。还通过利用 3D 流水线的功能性由于具有更好缩放和滤波而改进了伸拉位图引擎。而且，3D 流水线的高度流水格式被有利地用来提高伸拉位图引擎的性能。

实现对标准 3D 流水线的很多变化。修改设置引擎和窗口器以使 3D 流水线能处理除三角形之外的矩形。另外，修改映射地址产生器以便能够在 4:2:2 和 4:2:0 数据中寻址。另外，修改高速缓存以便能够对这些类型的数据寻址。下面将更详细地讨论所有这些变化。

图 1 是本发明可以在其中实现的计算机系统的方框图。该系统

01.07.13

5 可包括一个或一个以上的处理器 110. 可把所述处理器 110 耦合到主机总线 115. 可把所述主机总线 115 耦合到主机桥接器 120, 以便把图形卡或图形子系统 130 耦合到处理器 110. 也可把存储器 125 耦合到主机桥接器 120. 可进一步把所述主机桥接器 120 耦合到系统总线 150 上, 如 PCI 总线 150. 可把 PCI 总线 150 耦合到 PCI 至 ISA 桥接器 160. 可把端口以及扩展 ISA 总线耦合到 PCI 至 ISA 桥接器 160.

10 图形加速器 130 包括用于对三维图像着色的 3D 引擎 140. 正如下面将要描述的, 3D 引擎 140 还可用于伸拉位图. 可把各种输入和输出装置耦合到图形卡 130, 如视频输入、显示器、电视机等. 图形卡 130 还可包括用于加速图形操作的局部内存 147. 另外, 中央单元 140、145 可包括局部高速缓冲存储器, 如下所述. 可把图形卡 130 经总线 135 或类似的连接装置耦合到主机桥接器 120 上.

15 图 2 是 3D 流水线的方框图. 3D 引擎 210 包括纹理流水线 215. 如本领域中众所周知的, 可把这种 3D 引擎用于三维着色. 如下面将要描述的, 还可把 3D 流水线用于伸拉位图. 在一个实施例中, 响应指定了矩形基元类型的、在这里称为 GFXPRIMITIVE 的具体命令(以下称为 GFX 矩形命令)而执行伸拉位图解码; 但是, 也可以使用其他命令名称和格式.

20 命令流控制器 220 连接成从外部信源例如处理器或缓冲器接收命令. 命令流控制器 220 对命令分析和解码以执行适当的控制功能. 如果收到的命令不是 GFX 矩形命令, 命令流控制器 220 把控制信号和数据传递给设置引擎 225. 命令流控制器 220 还控制关于非 GFX 矩形命令的内存管理、状态变量管理、二维操作等等.

25 在一个实施例中, 当命令流控制器 220 收到 GFX 矩形命令时, 把顶点信息送到设置引擎 225. 在一个实施例中, 命令流控制器 220 为设置引擎 225 提供矩形的左上顶点、右下顶点和左下顶点. 设置引擎 225 用这些顶点来计算平面方程的三项: 关于 X、关于 Y 以及原点的属性变化率.

01-07-13

另外, 当执行纹理映射时, 设置引擎 225 确定包含由顶点定义的三角形的有边界框。

设置引擎 225 从命令流控制器 220 接收目标位置的左上顶点、右下顶点和左下顶点。设置引擎 225 还接收定义原始尺寸, 在每个顶点处的第二组坐标。设置引擎 225 计算两组坐标的比值以确定伸拉的比例, 在尺寸上或者扩展或者缩小。目标坐标是相对窗口的, 而源的坐标是相对于源图像的左上角, 即左上角坐标总是 0, 0。注意, 设置引擎方程是为用于 3D 着色的三角形所做计算的子集, 因而, 不需要附加的硬件。设置引擎 225 还产生边缘, 定义矩形基元的有边界框。设置引擎 225 向窗口器 230 提供有边界框信息。

窗口器 230 产生目标平面的地址。窗口器 230 一般通过划出经过每对顶点的线来产生这些地址。顶端线 510 以上的区域被标为负的, 而顶端线 510 以下的区域被赋予正值。底线 520 以下区域被标为负的, 而底线 520 以上区域被赋予正值。在边缘线 530 以内的区域被赋予正值, 而在边缘线 530 以外的区域被标为负的。这称为边缘测试。这确定了显示的三角形 540。在图 5A 和 5B 中说明了这些。

对于伸拉位图, 使用了窗口器 230 的另一方面。窗口器 230 永远不显示在由设置引擎 225 传递给窗口器 230 的有边界框 590 以外的区域。因而, 迫使边缘测试总为正, 由此把该区域定义为有边界框 590 的整个区域。窗口器 230 在有边界框内迭代像素以产生由 GFX 矩形命令写入的数据的读和写地址。换言之, 总是传递三角形边缘方程, 这使窗口器 230 能处理整个矩形而不是停留在三角形边界上。这些读和写地址被传递给映射地址发生器。

窗口器 230 产生像素读写地址以把数据接入高速缓冲存储器, 在图 2 中未示出。窗口器 230 还向色彩计算器 280 提供用于阿尔法( $\alpha$ )混合步骤的目标项。窗口器 230 从帧缓冲器获得目标信息。

窗口器 230 利用在 3D 流水线中可获得的深层流水线作业。当窗口器 230 从设置引擎 225 接收了有边界框和顶点时, 窗口器 230 开

01.07.13

始产生地址。用于这些地址的数据是从存储器中取出的。存储器具有相对长的延迟。因而，当窗口器 230 产生地址时，请求对数据进行流水作业，以 FIFO(先进先出)方式设置每个请求的控制部分。因而，对用于每个产生的地址的数据的请求随着地址产生而连续地被送往存储器。因而，存储器延迟仅影响所请求的第一像素数据。通过流水线作业，消除了存储器延迟对随后的像素数据的影响，并且可以按每一个时钟脉冲产生像素，维持通过量。流水线作业就面积而论是昂贵的，而把现有的流水线作业再用于 3D 引擎是有利的。

映射地址产生器 240 产生实际地址以在纹理映射中提取特定的像素。在一个实施例中，映射地址发生器 240 按照以下方程计算目标区域：

$$u(x, y) = \frac{C_{sx} \cdot x + C_{sy} \cdot y + C_{so}}{C_{sw} \cdot x + C_{yw} \cdot y + C_{ow}} + P_{ox}$$

这个方程对于那些熟悉 3D 着色的人是众所周知的。

通过使用下表中的值，复杂的纹理映射方程可以被简化以用于伸拉位图计算，由此使硬件能用于两个目的。C<sub>so</sub> 和 C<sub>ow</sub> 被直接发送给映射地址产生器 240 作为平面系数。

变量	说明	值
C <sub>so</sub>	S(源平面中的水平尺寸)相对于 x 的变化率	由设置引擎计算
C <sub>ow</sub>	对 S 的偏移	由设置引擎计算
C <sub>sy</sub>	S 相对于 y 的变化率	由设置引擎计算
C <sub>st</sub>	T 相对于 x 的变化率	由设置引擎计算
C <sub>ot</sub>	对 T 的偏移	由设置引擎计算
C <sub>yt</sub>	T 相对于 y 的变化率	由设置引擎计算
C <sub>xsw</sub>	1/W 相对于 x 的变化率	设置为 0.0
C <sub>oyw</sub>	对 1/W 的偏移	设置为 1.0
C <sub>yiw</sub>	1/W 相对于 y 的变化率	设置为 0.0

01-07-13

$P_{ou}$	对 u 的偏移	由设置引擎计算, 但通常设置为 0.0
$P_{ov}$	对 v 的偏移	由设置引擎计算, 但通常设置为 0.0

这迫使在上述方程中的分母为一。这产生简化的方程:

$$u(x, y) = C_{ox} \cdot x + C_{oy} \cdot y + C_{os}$$

这个方程定义了一个简单平面  $Ax+By+C$ . 进一步修改映射地址产生器 240 以接受对于顶点的非归一化值。视频地址一般都不是归一化的, 而 3D 着色坐标组一般是归一化的。通过修改映射地址产生器 240 可操作的值, 使 3D 流水线可用于伸拉位图。

读取单元 245 执行存储器管理和串行化。读取单元 245 请求存储在高速缓存 260 或存储器中的源映射。读取单元 245 把由映射地址产生器 240 提供的读取地址转换成高速缓存地址。由读取单元 245 产生的高速缓存地址被送往高速缓存 260. 如果在高速缓存 260 中未发现数据, 读取单元 245 向存储器请求数据, 并且当数据到达时把它放入高速缓存。

存储在高速缓存地址中的像素数据被送往滤波器 250. 映射地址产生器 240 把部分像素定位数据和关于相邻像素的高速缓存地址发送到滤波器 250. 在本领域中滤波技术是众所周知的, 这里不再进一步讨论。

内插器 270 可用于产生用于混合的内插  $\alpha$  值。 $\alpha$  混合使部分已经在帧缓冲器中的图像和部分源图像能被看到。术语  $\alpha$  定义源的不透明度。对于一个实施例,  $\alpha$  值对于伸拉位图是固定的, 并且未使用内插器 270. 如果  $\alpha$  值不固定, 则使用如上所述关于映射地址产生器 240 的同样方程。对于另一个实施例, 指定所有三个顶点的  $\alpha$  值, 并且内插器 270 产生在各顶点的  $\alpha$  值之间的光滑混合。 $\alpha$  值被传递到色彩计算器 280.

01-07-13

在下表中说明彩色空间转换和通道分配格式的一个实施例。

源 → 目标 表面	操作	源表面	目标表面
4:2:0 → 4:2:0	动作补偿和 算法伸拉位 图	分开处理 4:2:0 表面的每个通 道, 需要 3 遍	4:2:0
4:2:0 → 4:2:2	算法伸拉位 图	4:2:2	分开处理 4:2:0 表面的每个通 道, 需要 3 遍
4:2:2 → 4:2:2	算法伸拉位 图	4:2:2	4:2:2
4:2:2 → 4:2:0	算法伸拉位 图	4:2:2	分开处理 4:2:0 表面的每个通 道, 需要 3 遍
4:2:2 → 16bpp RGB	算法伸拉位 图	4:2:2 带有彩色 空间转换	16bpp 表面
8bptY → 16bpp RGB	利用纹理合 成的 3D 着 色	亮度纹理映射 调制另一映射	16bpp 反向缓冲
16bpt AI → 16bpp RGB	利用纹理合 成的 3D 着 色	$\alpha$ 及强度纹理 映射混合和/或 调制另一映射	16bpp 反向缓冲
8 bpt 索引 RGB → 16bpp RGB	3D 着色	8 比特索引用于 调色纹理映射	16bpp 反向缓冲
16bpt RGB → 16bpp RGB	3D 着色	16bpt 纹理映射	16bpp 反向缓冲

注: bpp - 每像素位数, bpt -

对于另一个实施例, 可从支持的源表面和输出方式中产生其他

01-07-13

目标表面。

通道操作单元 255 执行颜色通道分配操作(Y、U 和 V 通道)。它还执行从 YUV 到 RGB 的彩色空间转换。对于另一个实施例，通道操作单元 255 可以既进行通道分配、又进行彩色空间转换。对于一个实施例，对 YUV 通道转换要用三遍。首先，每个像素被上抽样(up sampled)为 4:4:4 格式，对中间像素内插 U 和 V 值。然后，可能发生三遍表面格式转换。在第一遍中，写入 Y 值，在第二遍中写入 U 值，而在第三遍中写入 V 值。这提供了从平面到打包格式的便宜转换，并且不需要三个地址流式磁带机。这些转换可能用于电视会议和其他用途。

对于一个实施例，在色彩计算器 280 的数据校准寄存器中进行从 4:2:0 到 4:2:2 的通道转换。

对于一个实施例，色彩计算器 280 尤其是使用以下混合方程：

$$\text{色彩} = \alpha * \text{源} + (1 - \alpha) * \text{目标}$$

色彩计算器 280 接收三个输入值：来自通道操作单元 255 的源色彩；由窗口器 230 取出的、来自存储器的目标色彩；以及来自内插器 270 的  $\alpha$  值(它可能是固定的)。在色彩计算器 280 中的打包逻辑(packing logic)对数据进行校准并且把它写到目标平面。

图 3 是图像伸拉过程的流程图。在块 310 中，启动矩形基元类型。矩形基元类型使 3D 引擎能识别由三个点描述为矩形的对象。所述矩形是由按照与启动的淘汰顺序(culling order)一致的正确提升顺序(winding order)的三个顶点描述的轴校准的矩形。这些顶点描述了一个直角三角形，其中三角形的底边与 X 轴平行而三角形的垂直边与 Y 轴平行。基于这三个顶点而知道第四点。对于一个实施例，如果当开始伸拉位图时已经启动矩形基元类型，这个步骤可以删去。

在块 320，矩形命令被送到 3D 引擎，矩形命令向 3D 引擎指出以下信息将被伸拉位图，而不是三维着色信息。

在块 330，该过程测试 3D 流水线是否被刷新。对于一个实施例，

01-07-13

5 因为 3D 流水线允许并行操作，所以，在开始视频伸拉之前它被刷新。对于一个实施例，这一步骤和步骤 335 可被删去。如果 3D 流水线未被刷新，则过程继续到块 335，其中过程等待直到 3D 流水线被刷新。然后该过程继续到块 340。如果 3D 流水线已被刷新，则过程直接继续到块 340。

在块 340，矩形数据被送到 3D 流水线 210 中的设置引擎 225。矩形数据定义所操作的数据的矩形。

在块 345，在窗口器中计算 XY 坐标。XY 坐标定义图象目标表面。

10 在块 350，在映射地址产生器(MAG)中计算 UV 坐标。U 和 V 坐标定义图像源平面。

在块 360，提取像素数据并将其高速缓存。对于一个实施例，这在产生像素地址后发生。对于一个实施例，像素数据以 4X4 数据块被高速缓存。

15 在块 370，内插像素值。这产生在目标表面上每个像素的属性。

在块 380，执行通道分配和彩色空间转换步骤。对于一个实施例，仅在被请求时才执行这些步骤。对于另一实施例，总是执行这些步骤。

20 在块 390，色彩计算器对数据进行  $\alpha$  混合、校准数据并把它写到目标表面。这产生目标图像。在这个过程的结尾，在目标图像中的每个像素都是已知的，并且显示出目标图像。

在前述的说明书中，已经参照其特定的示例性实施例描述了本发明。但是，显然，只要不违背如所附权利要求书中所陈述的本发明的更广的精神和范围，可以做各种修改和变化。因此，说明书和附图是为了说明而非为了限定。

01-07-13

## 说 明 书 附 图

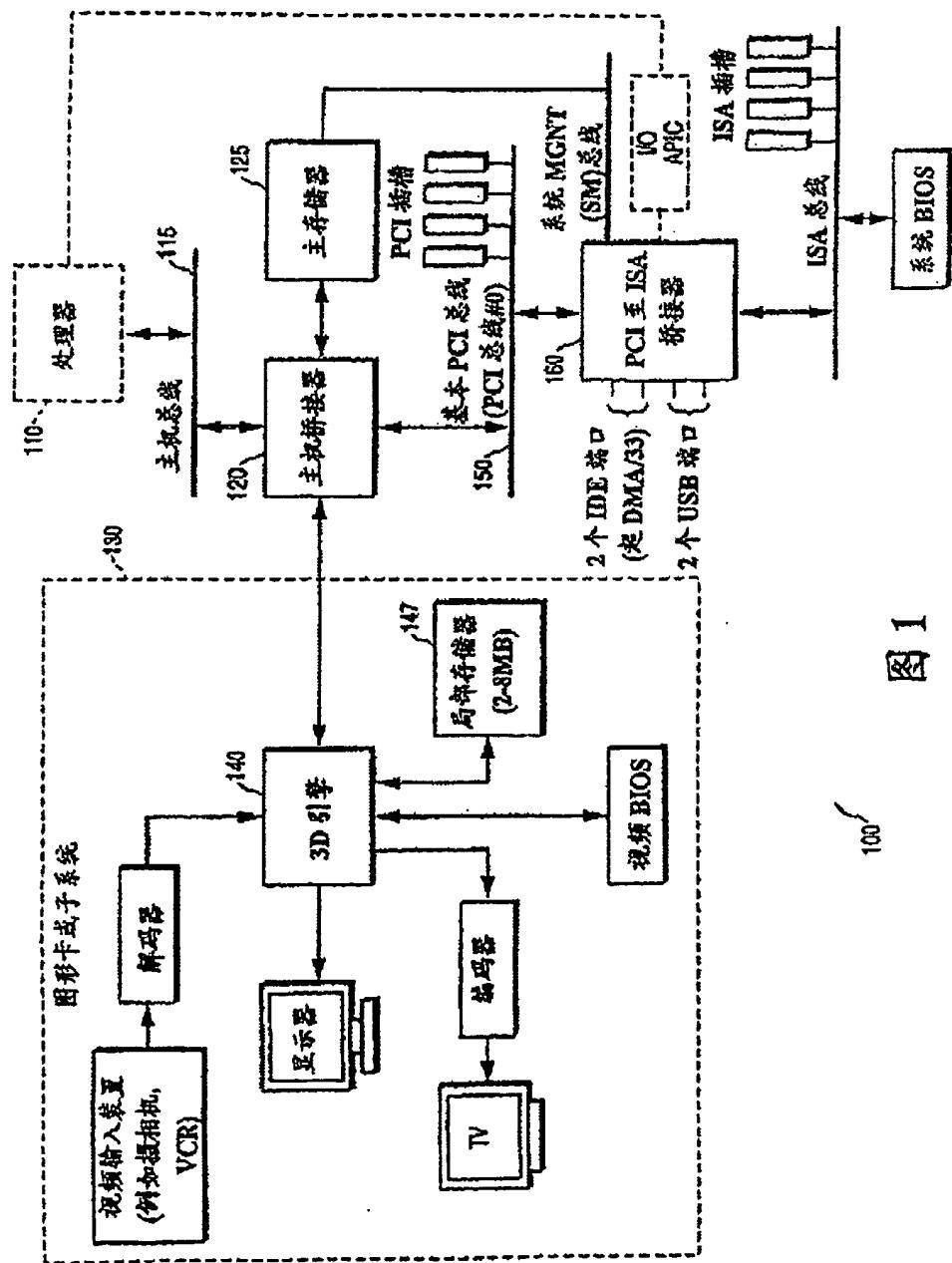
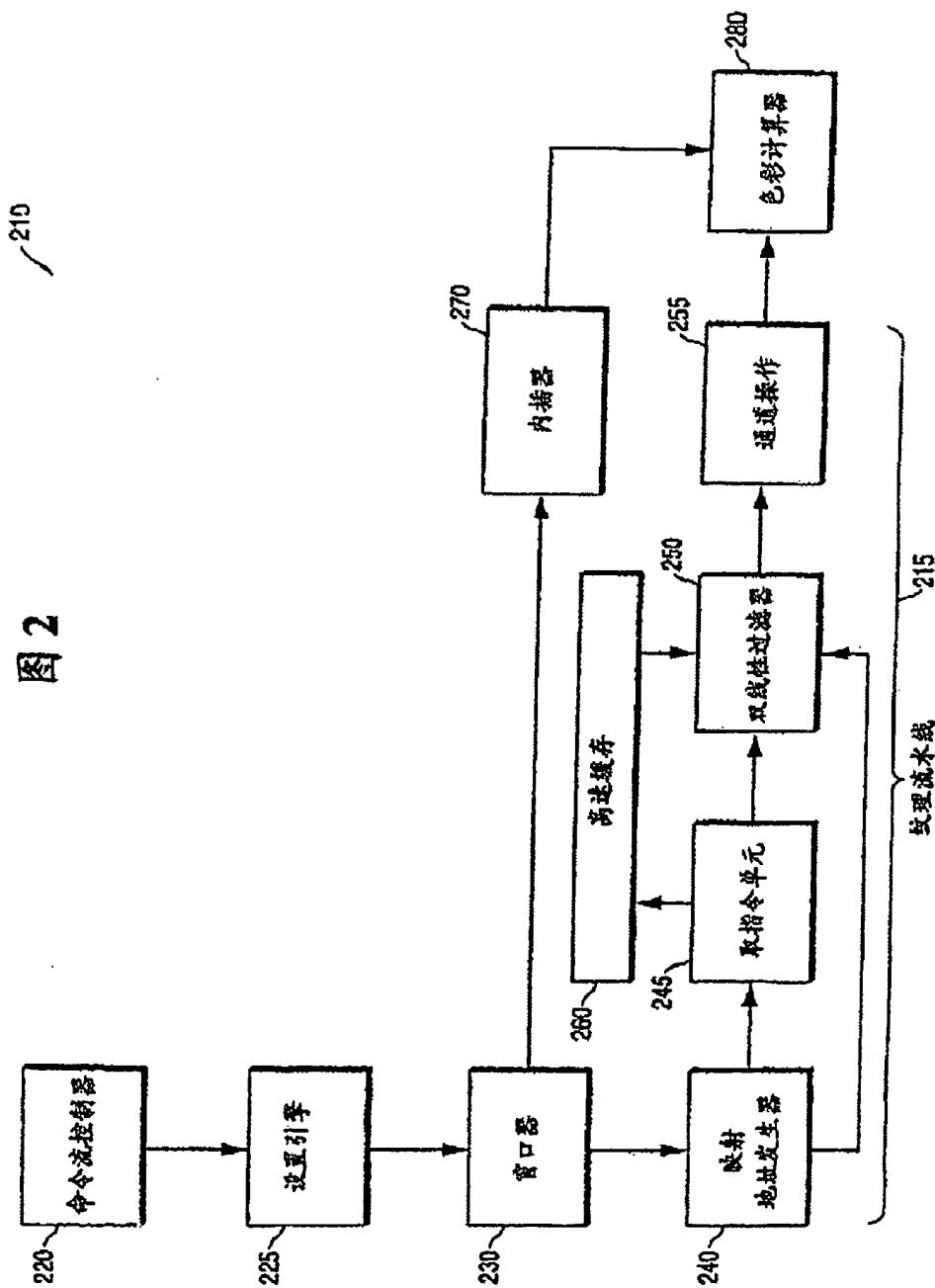


图 1

01-07-13

图 2



01-07-13

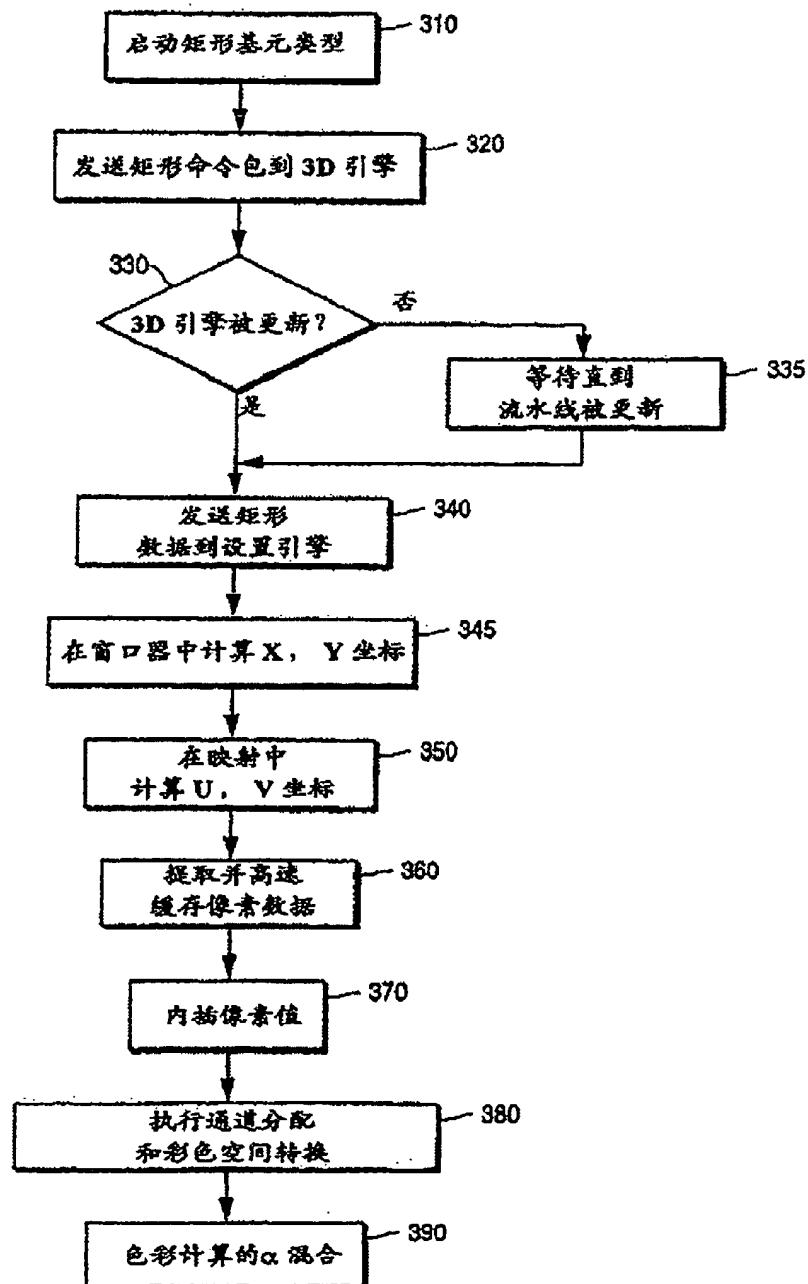


图 3

- 3 -

01-07-13

400

410				420				430				440			
0	1	2	3	16	17	18	19	32...				48...			
4	5	6	7	20	21	22	23								
8	9	10	11	24	25	26	27								
12	13	14	15	28	29	30	31								
450				460				470				480			

图 4

01-07-13

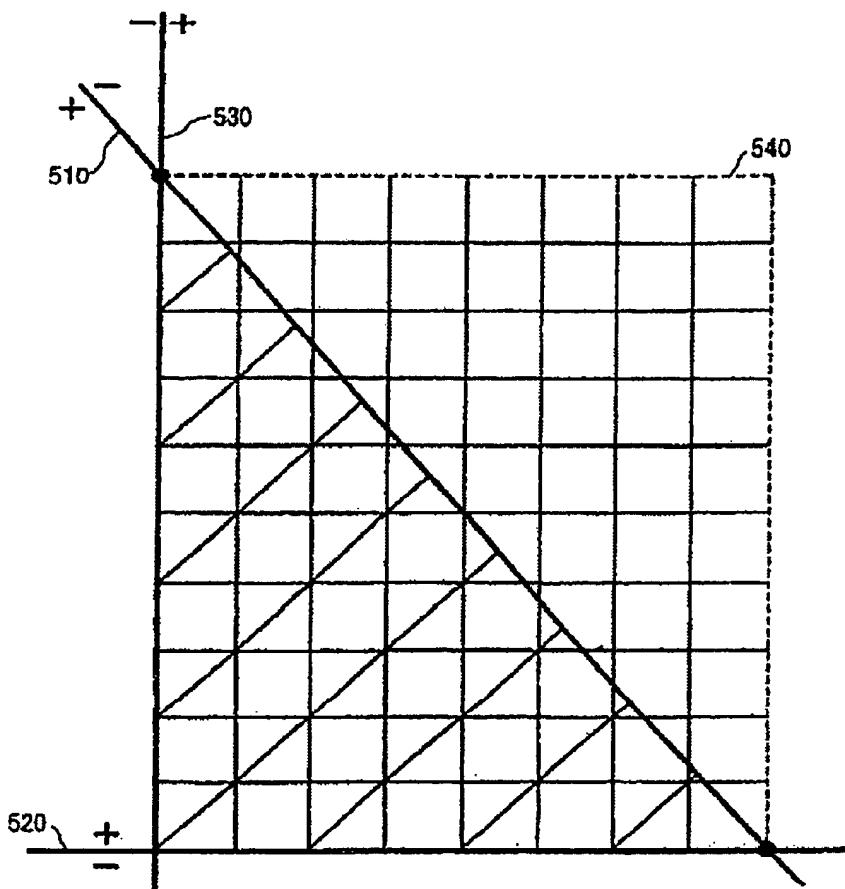


图 5A

01-07-13

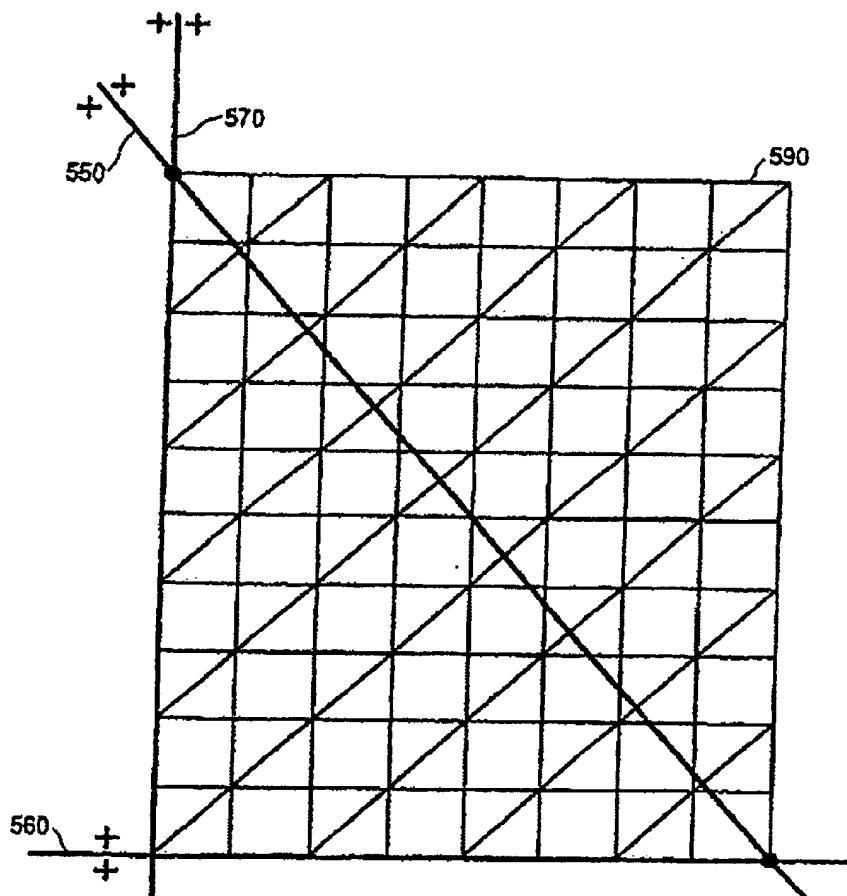


图 5B

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**